

基础知识 - - JAVA成员内部类 PDF转换可能丢失图片或格式  
， 建议阅读原文

[https://www.100test.com/kao\\_ti2020/278/2021\\_2022\\_\\_E5\\_9F\\_BA\\_E7\\_A1\\_80\\_E7\\_9F\\_A5\\_E8\\_c104\\_278570.htm](https://www.100test.com/kao_ti2020/278/2021_2022__E5_9F_BA_E7_A1_80_E7_9F_A5_E8_c104_278570.htm) 作为外部类的一个

成员存在，与外部类的属性、方法并列。 publicclass Outer {  
privatstaticinti = 1. privateintj = 10. privateintk = 20.

publicstaticvoidouter\_f1() { } publicvoidouter\_f2() { } // 成员内部  
类中，不能定义静态成员 // 成员内部类中，可以访问外部类的

的所有成员 class Inner { // static int inner\_i = 100.//内部类中不允  
许定义静态变量 intj = 100. // 内部类和外部类的实例变量可以

共存 intinner\_i = 1. void inner\_f1() { System.out.println(i). //在内  
部类中访问内部类自己的变量直接用变量名

System.out.println(j). //在内部类中访问内部类自己的变量也可  
以用this.变量名 System.out.println(this.j). //在内部类中访问外

部类中与内部类同名的实例变量用外部类名.this.变量名

System.out.println(Outer.this.j). //如果内部类中没有与外部类同  
名的变量，则可以直接用变量名访问外部类变量

System.out.println(k). outer\_f1(). outer\_f2(). } } //外部类的非静

态方法访问成员内部类 publicvoidouter\_f3() { Inner inner = new  
Inner(). inner.inner\_f1(). } // 外部类的静态方法访问成员内部类

，与在外部类外部访问成员内部类一样

publicstaticvoidouter\_f4() { //step1 建立外部类对象 Outer out =  
new Outer(). //step2 根据外部类对象建立内部类对象 Inner

inner = out.new Inner(). //step3 访问内部类的方法

inner.inner\_f1(). } publicstaticvoid main(String[] args) {

//outer\_f4().//该语句的输出结果和下面三条语句的输出结果一

样 //如果要直接创建内部类的对象，不能想当然地认为只需加上外围类Outer的名字，//就可以按照通常的样子生成内部类的对象，而是必须使用此外围类的一个对象来//创建其内部类的一个对象：`//Outer.Inner outin = out.new Inner()` //因此，除非你已经有了外围类的一个对象，否则不可能生成内部类的对象。因为此//内部类的对象会悄悄地链接到创建它的外围类的对象。如果你用的是静态的内部类，//那就不需要对其外围类对象的引用。`Outer out = new Outer(). Outer.Inner outin = out.new Inner(). outin.inner_f1(). }` } } 注意：内部类是一个编译时的概念，一旦编译成功，就会成为完全不同的两类。对于一个名为outer的外部类和其内部定义的名为inner的内部类。编译完成后出现outer.class和outer\$inner.class两类。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)